

FIG. 1

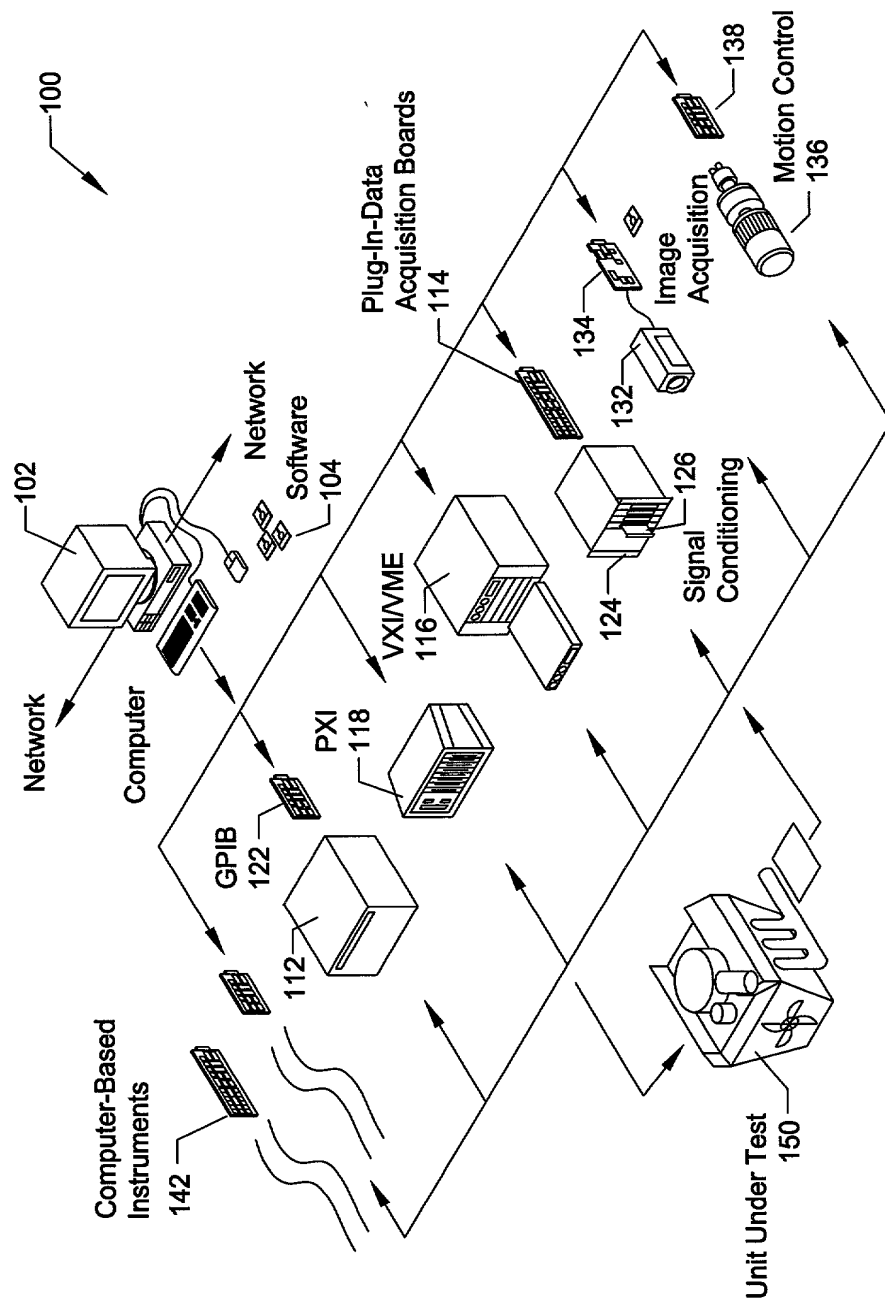


FIG. 2A

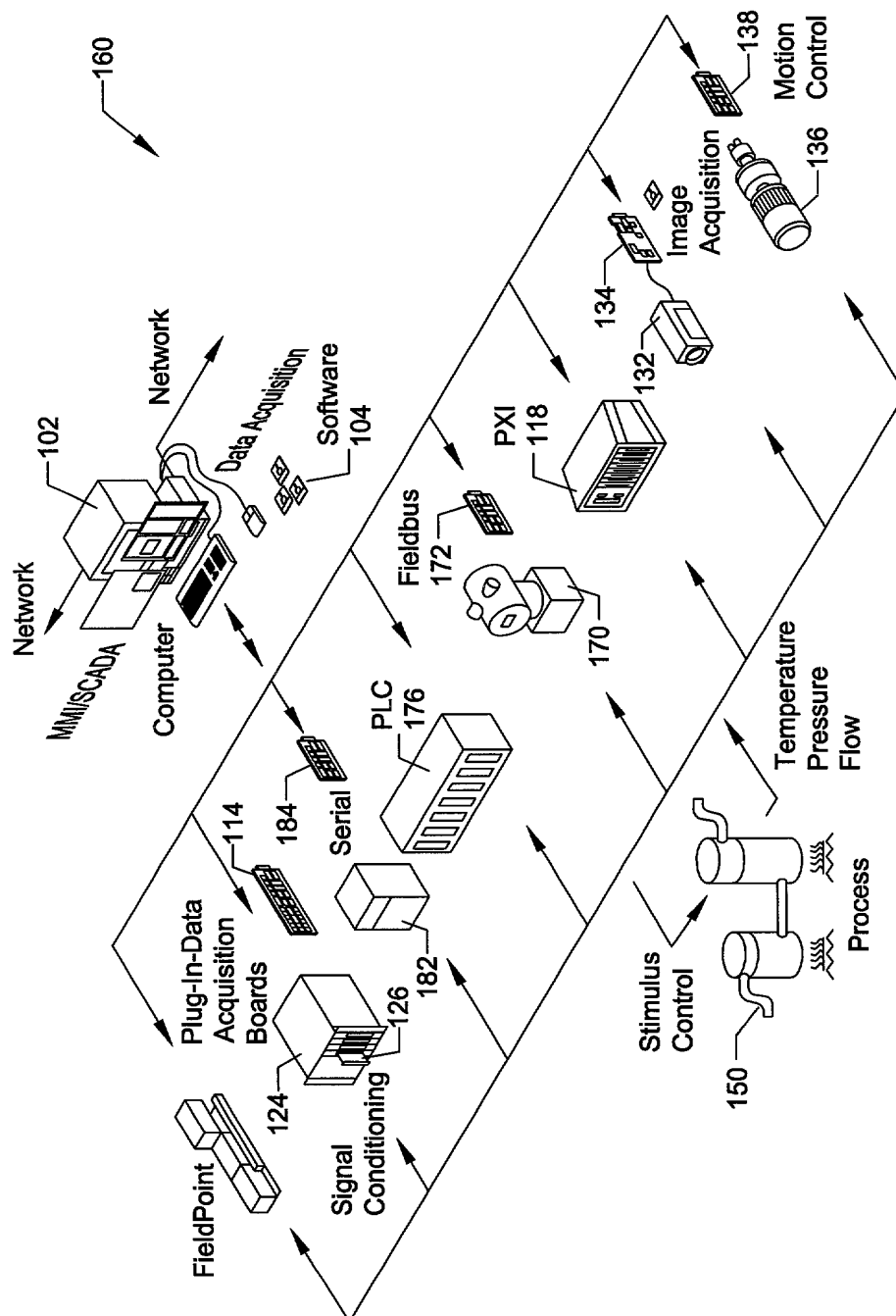


FIG. 2B

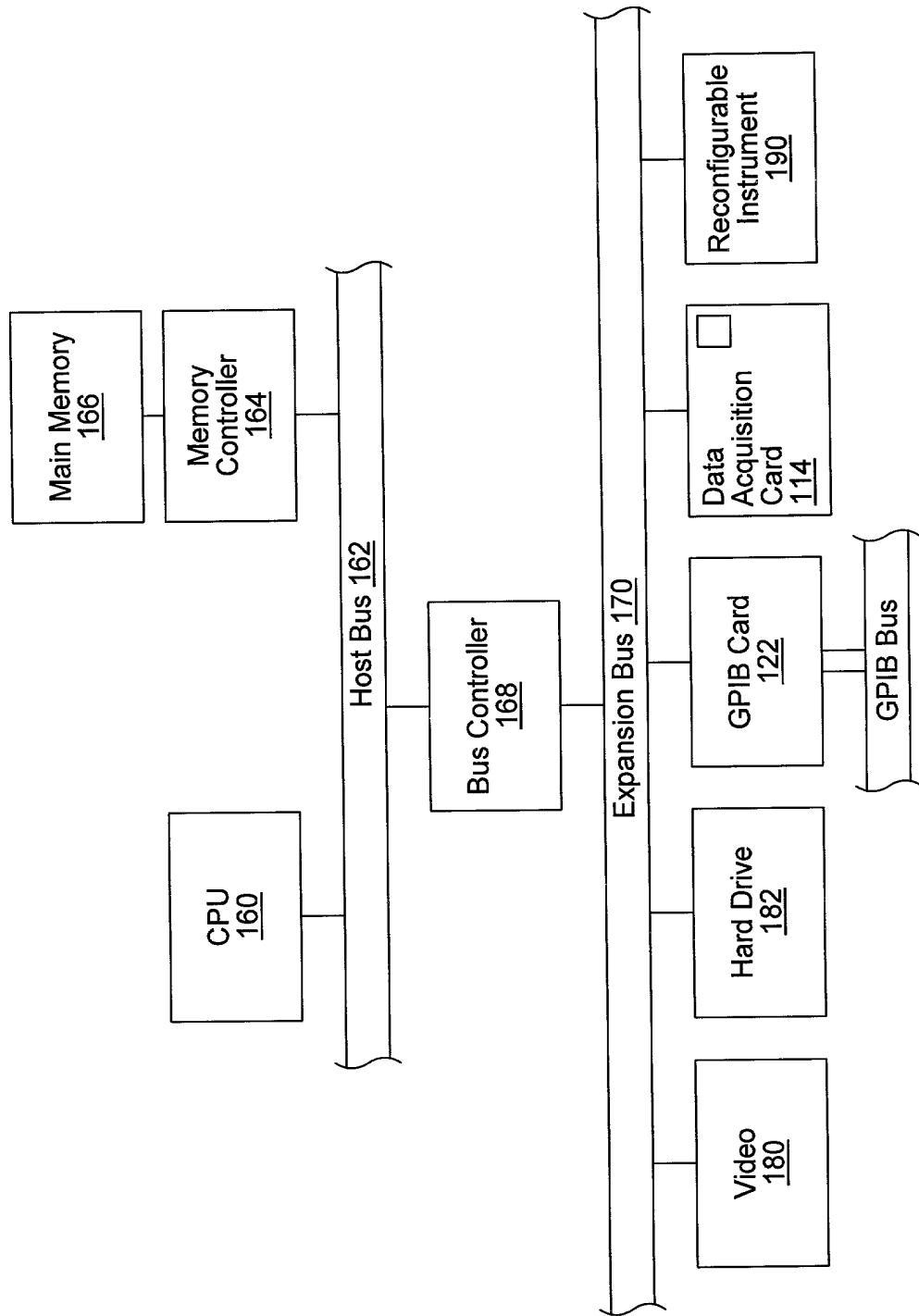


FIG. 3

Developer creates a graphical program generation (GPG) program, wherein the GPG program is operable to generate a plurality of graphical programs, based on received information
200

Specify program information, e.g., in response to user input, wherein the program information specifies desired functionality to be implemented in a graphical program
204

execute graphical program generation (GPG) program
206

GPG program receives information specifying functionality for a graphical program (or graphical program portion)
208

GPG program programmatically generates a graphical program (or graphical program portion) to implement the specified functionality
210

FIG. 4

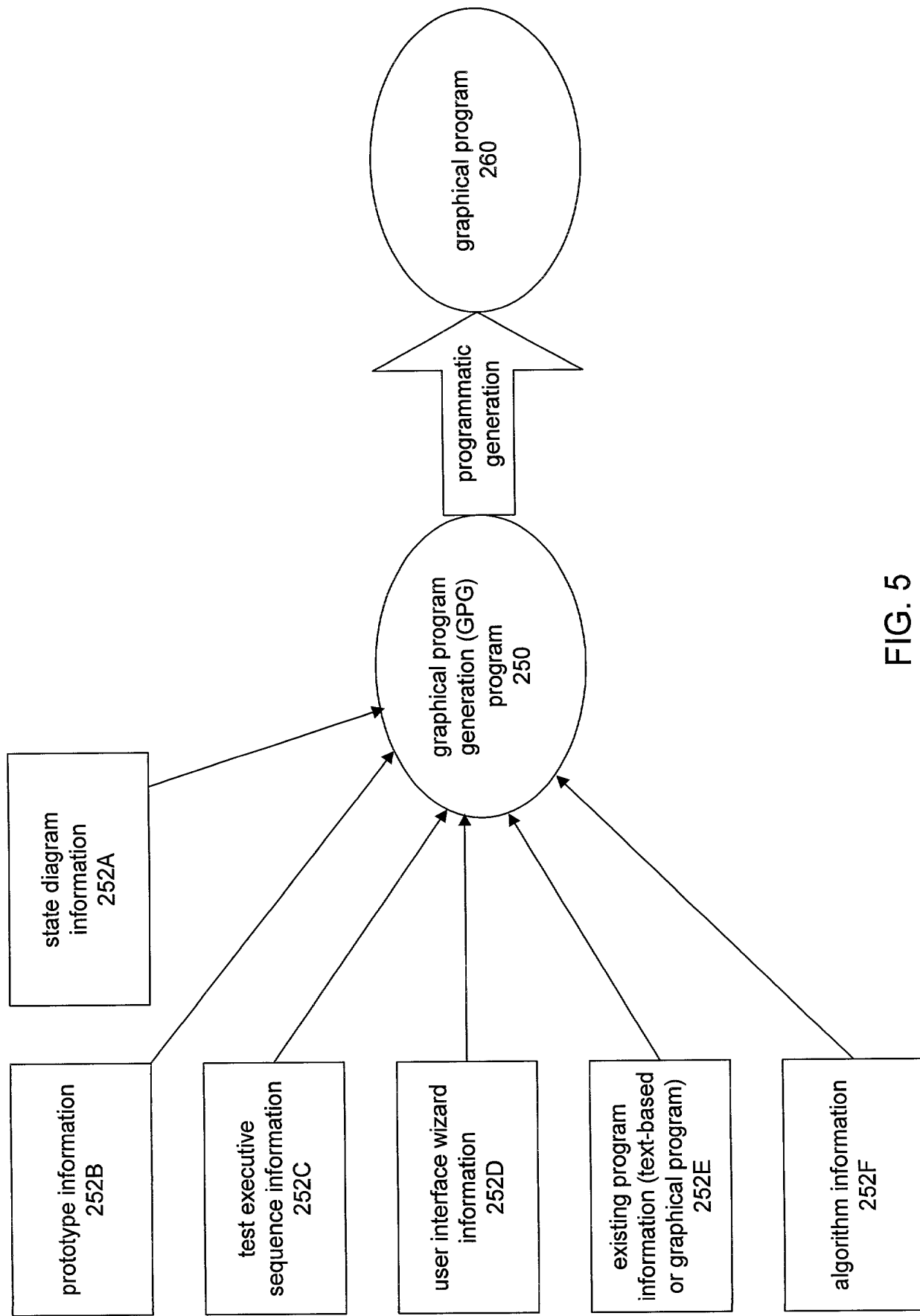


FIG. 5

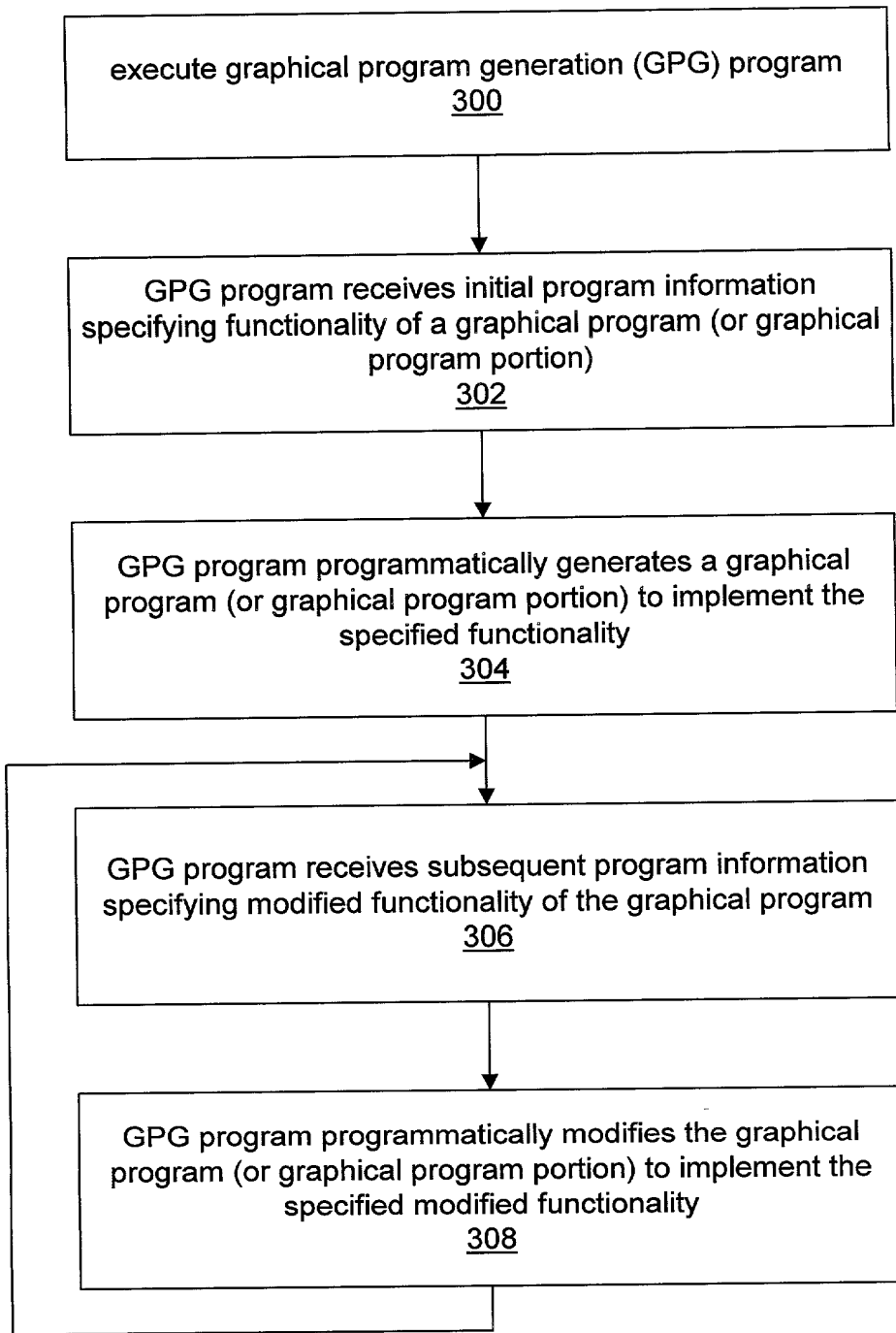


FIG. 6

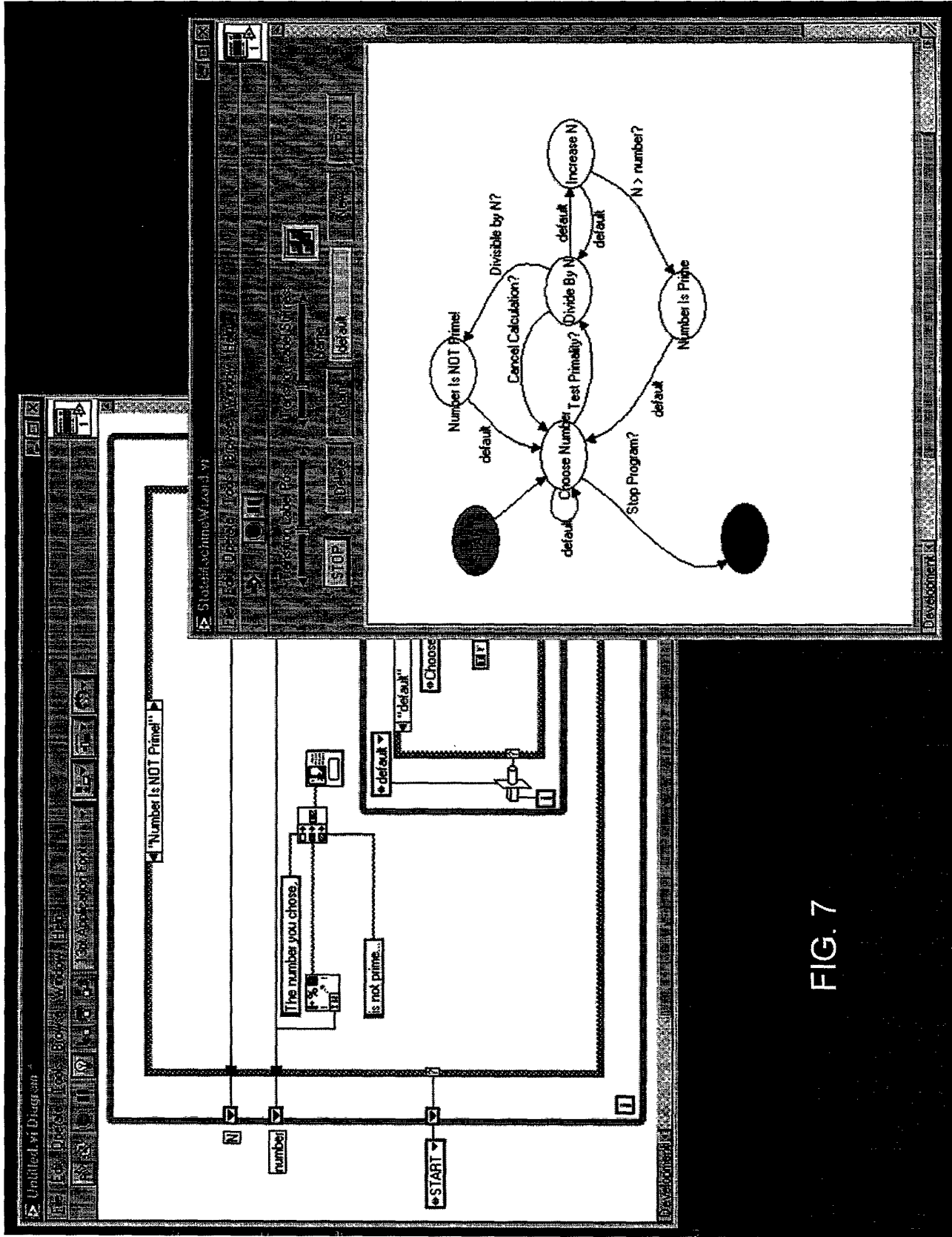


FIG. 7

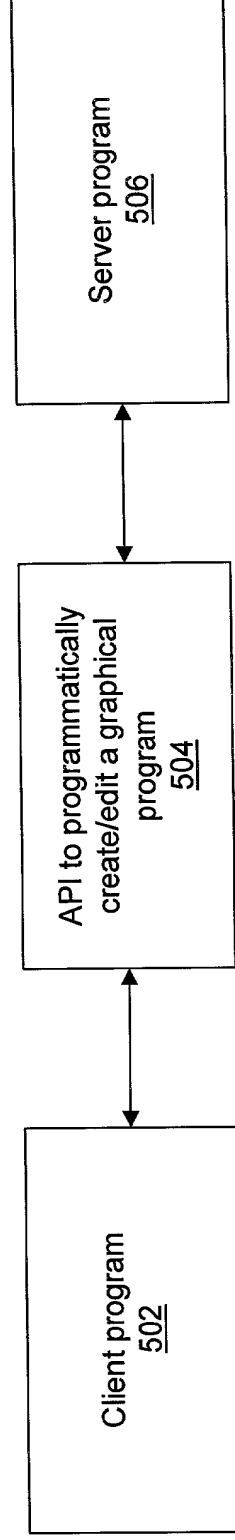


FIG. 8

Diagram illustrating the VI (Value Input) block structure and its inputs/outputs:

- Inputs (Left):**
 - type specifier
 - application reference (local)
 - VI name or path
 - error in (no error)
 - password
- Outputs (Right):**
 - application reference
 - error out
- Block Labels:** VI (top), O (bottom)
- Internal Structure:** A 2x2 grid of squares, with the top-left square filled black.

000221 625460

New VI Reference Node

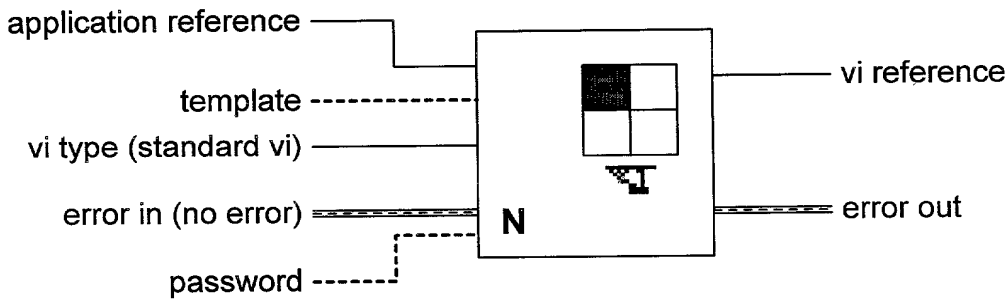


FIG. 11

Open VI Object Reference Node

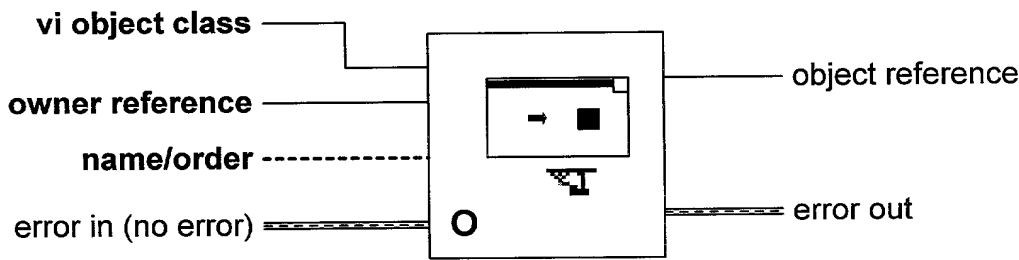


FIG. 12

New VI Object Reference Node

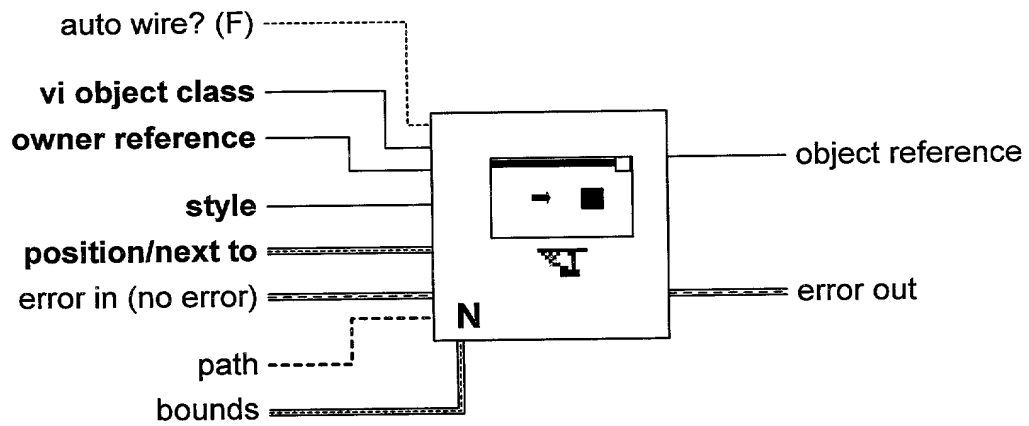


FIG. 13

Upcast Reference Node

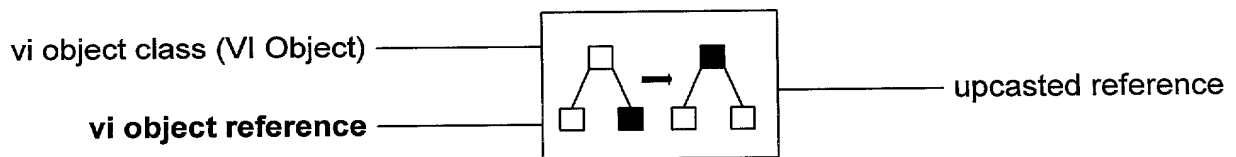


FIG. 14

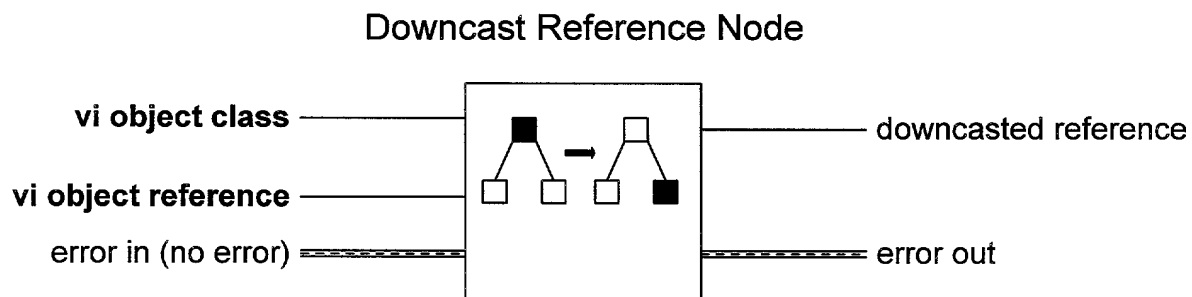


FIG. 15

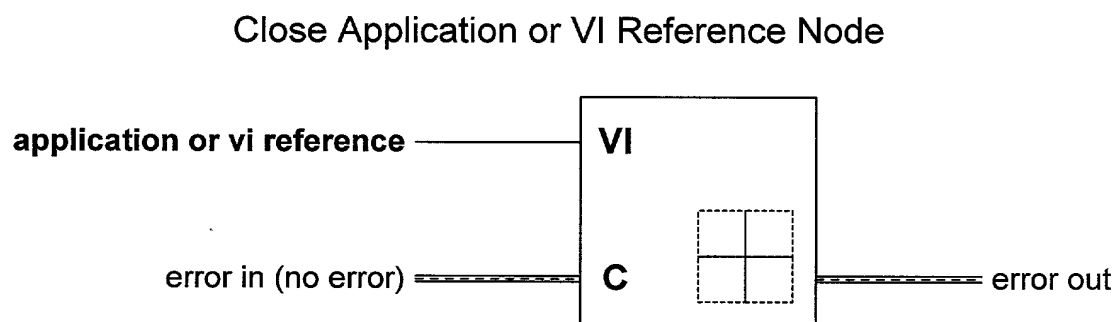


FIG. 16

Call By Reference Node

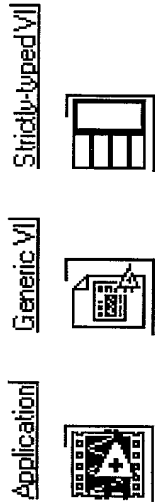


FIG. 20

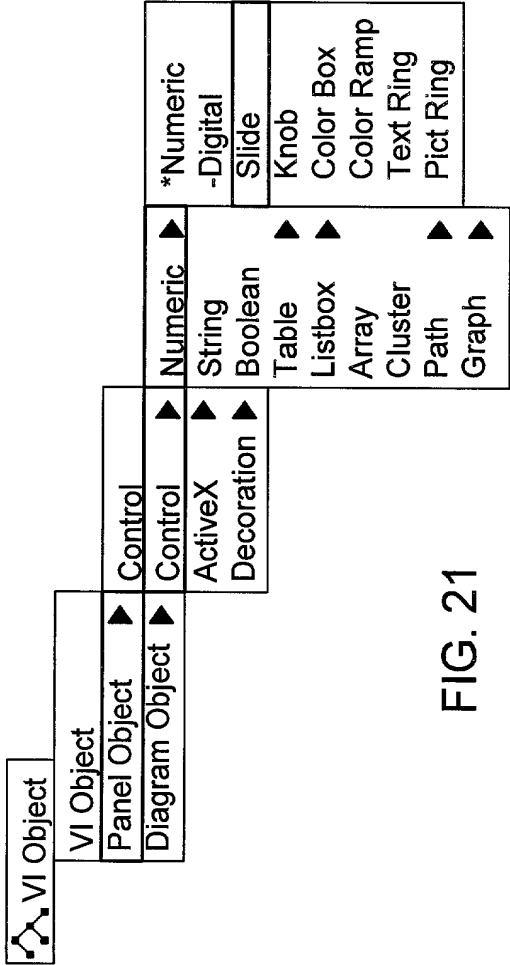


FIG. 21

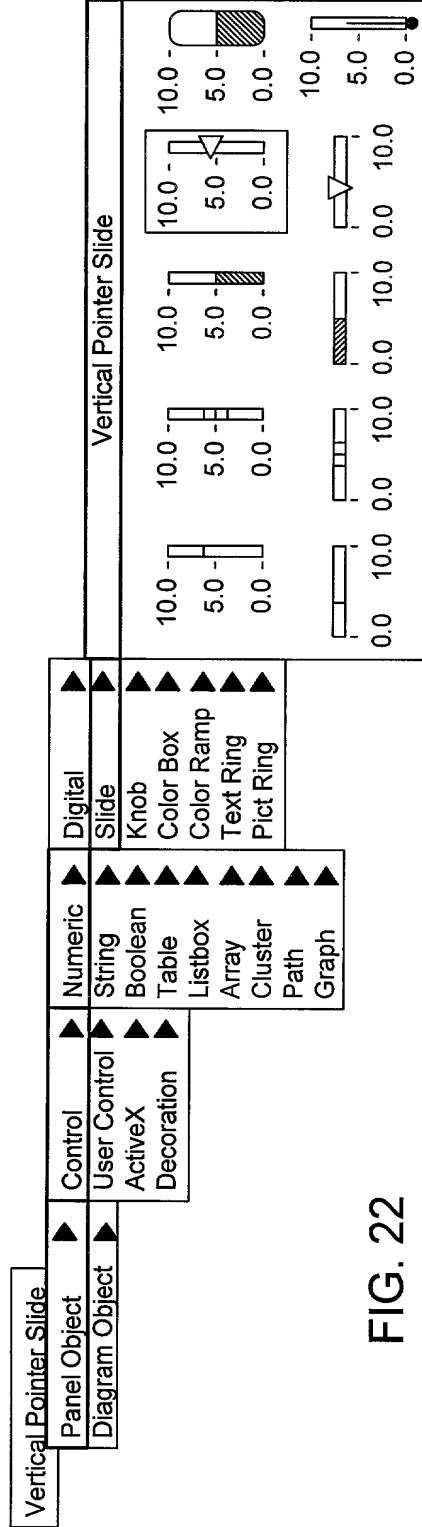


FIG. 22

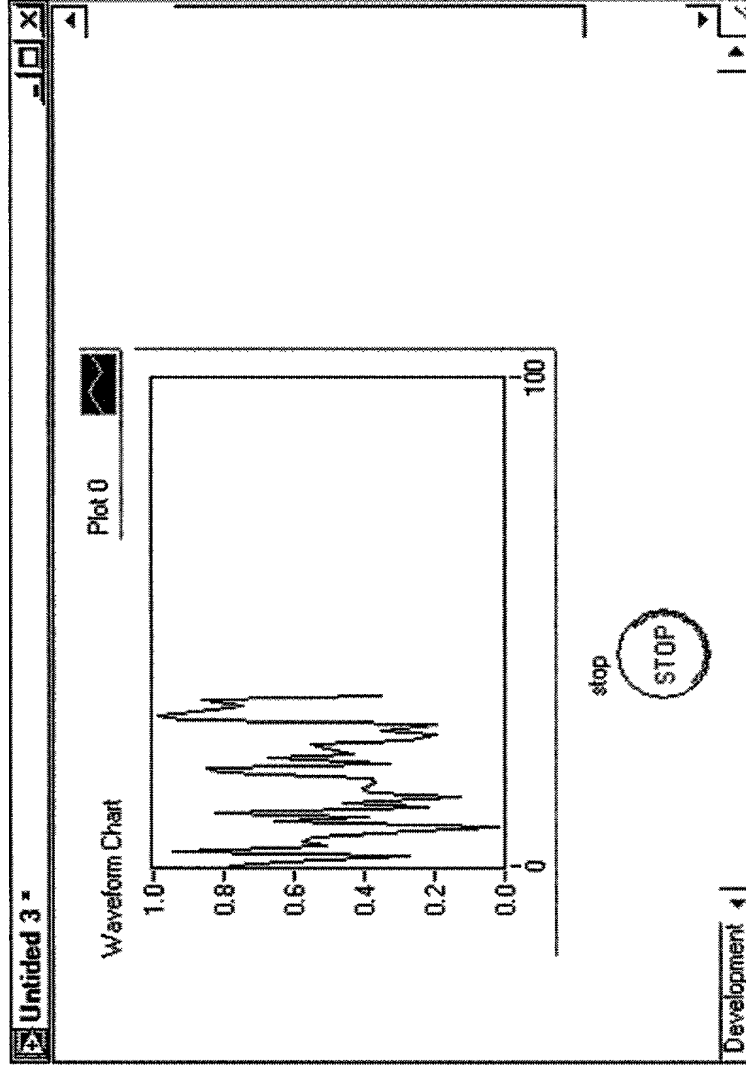


FIG. 23

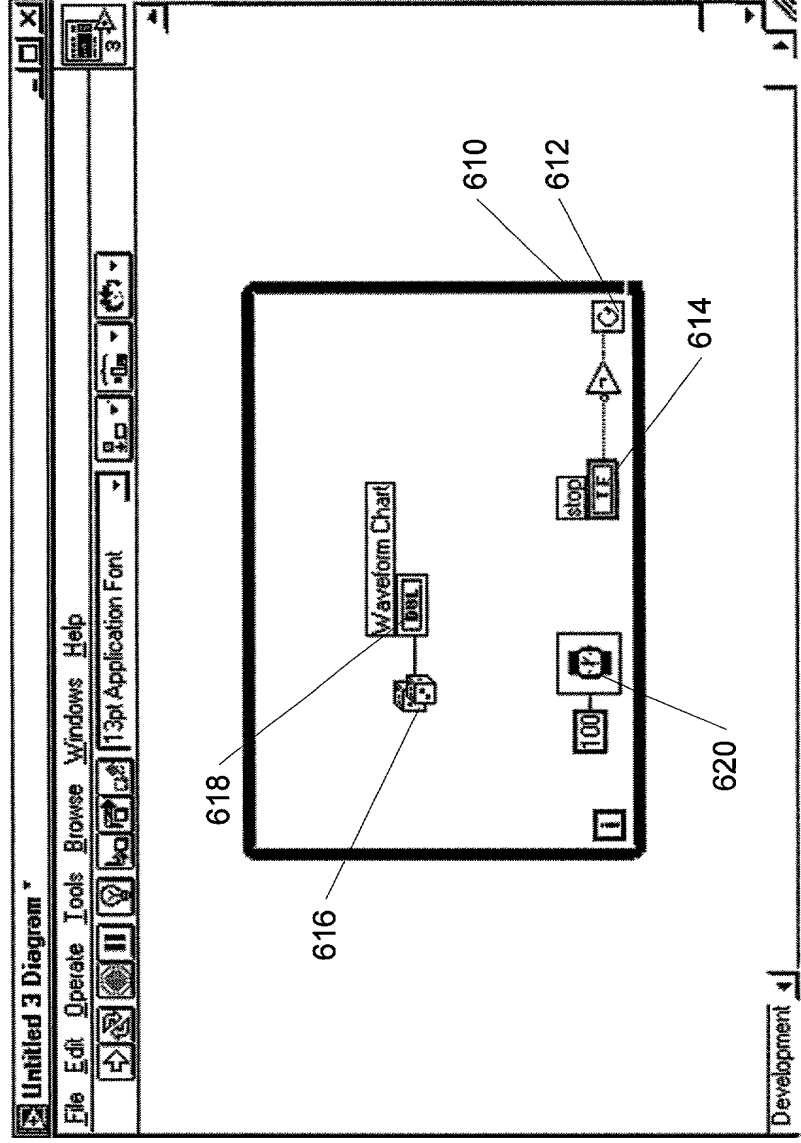


FIG. 24

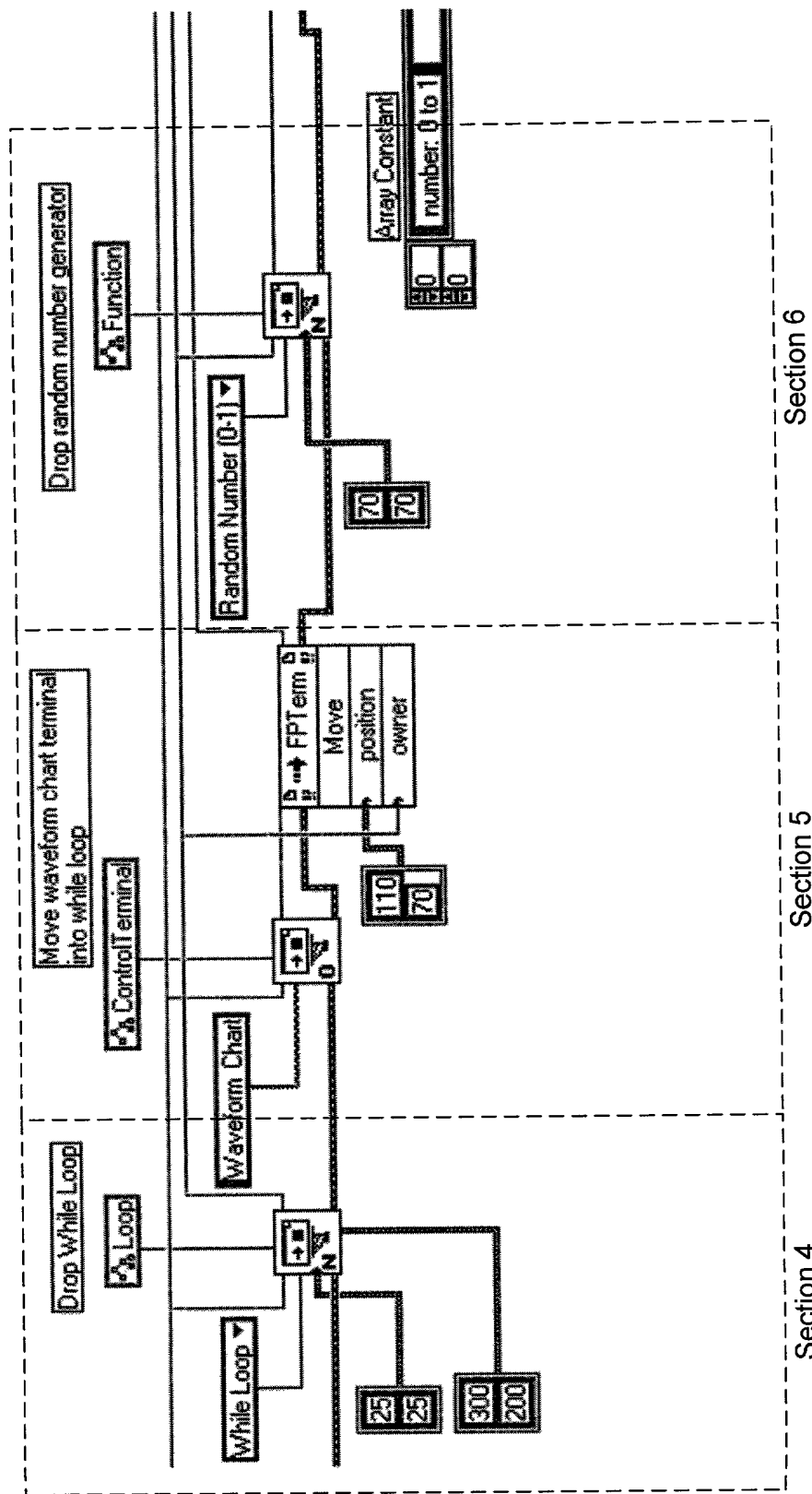


FIG.25B
(Continued)

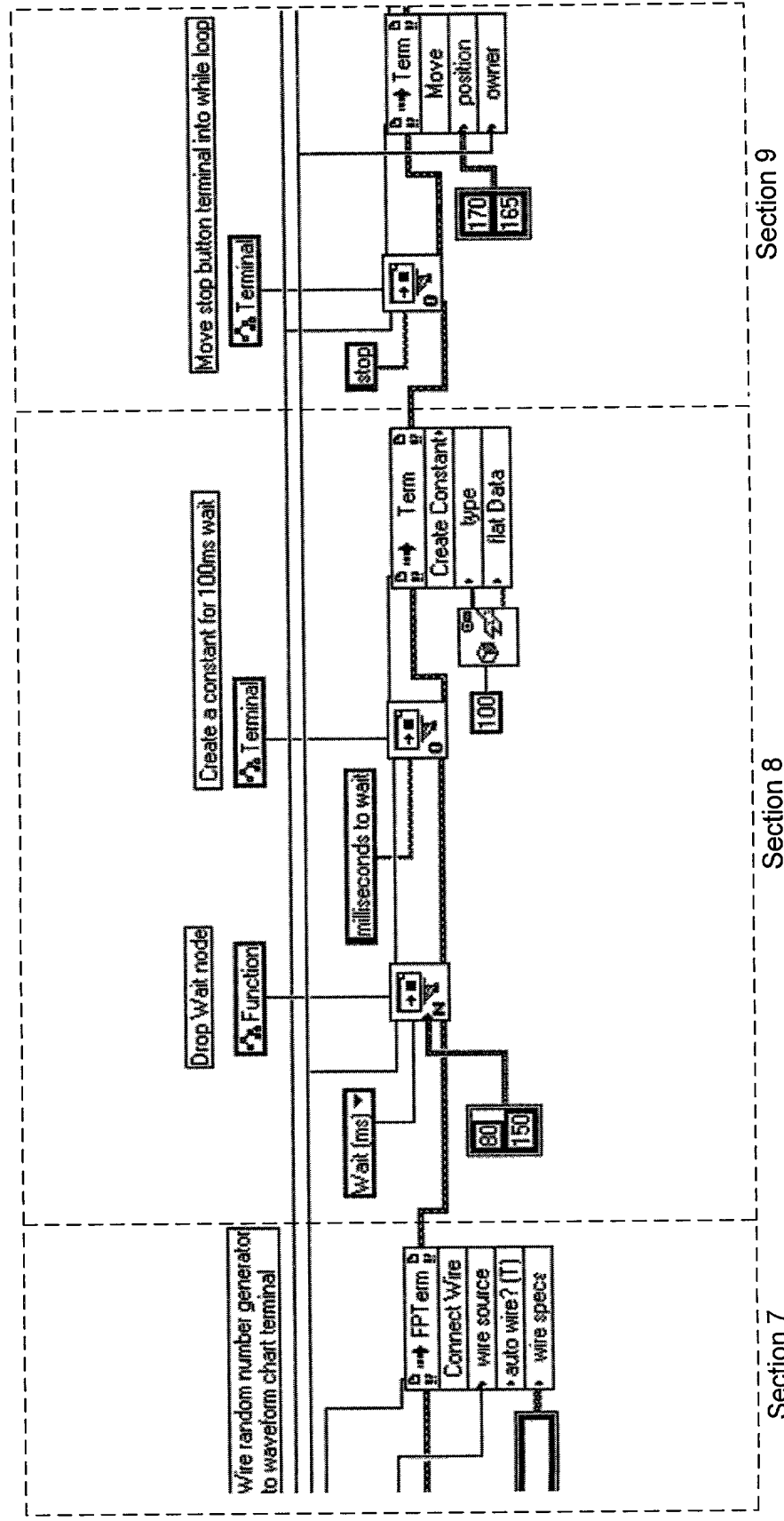


FIG.25C
(Continued)

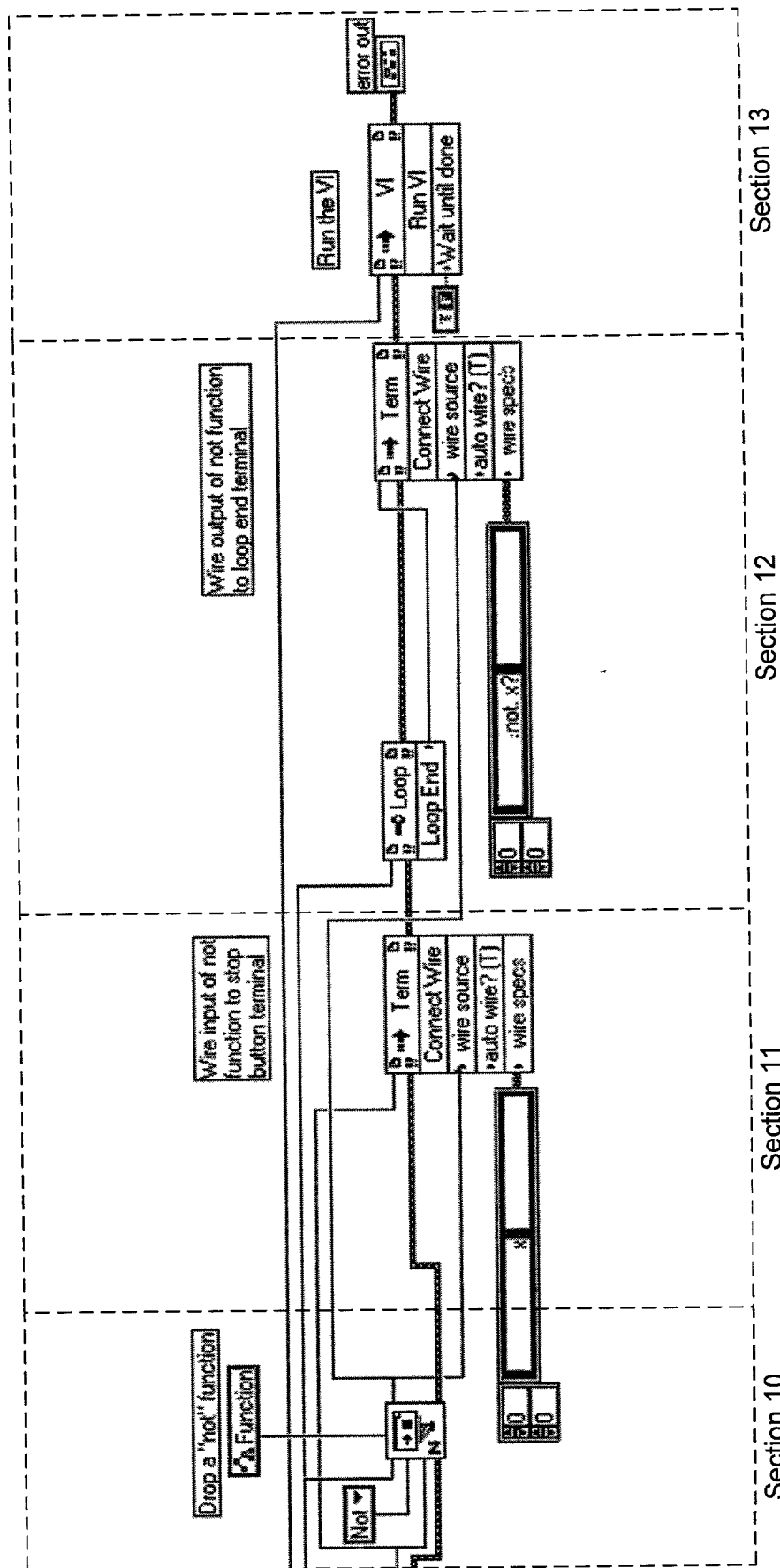


FIG. 25D
(Continued)